SauceLabs

SAUCE LABS × WAKEFIELD RESEARCH

The Enterprise AI Code Verification Crisis

AI in Software Testing 2026

What happens when organizations bet big on AI-generated code before they can verify it.

80%
traced an AI-code incident

66%
traded quality for speed

400
leaders surveyed

More code. Less certainty.

There's a version of this story that writes itself (and executives are already telling): AI floods the zone with code, teams ship faster, software quality holds, customers experience positive outcomes, everyone wins.

But when Sauce Labs tapped Wakefield Research to survey 400 of those executives and engineering leaders, a more complicated picture emerged. Leadership mandates AI adoption to move faster and do more with less. The code volume accelerates. But testing infrastructure — built for human-scale output — struggles to keep up, and defects slip through. Consequently, quality suffers, and the financial consequences of failures grow considerably.

New external research supports this survey's findings. A 2026 NBER working paper tracking more than 100,000 GitHub developers[1] found that AI coding tools drove a 741% increase in code output but only a 20% rise in actual software releases, a gap researchers attribute to human bottlenecks in review and testing. This report delves into what's actually happening inside that bottleneck.

CONTENTS

METHODOLOGY

An online survey of 400 U.S. business leaders was fielded by Wakefield Research for Sauce Labs in early 2026, exploring how the rapid adoption of agentic AI is reshaping software testing — and the roles, budgets, accountability, and trust surrounding it.

200
executives (C-suite)

200
engineering leaders

[1] Demirer, M., Musolff, L., & Yang, L. (2026). Writing code vs. shipping code: Productivity effects across generations of AI coding tools (NBER Working Paper No. 35275). National Bureau of Economic Research. https://doi.org/10.3386/w35275

SECTION 01

01

Coding Transparency

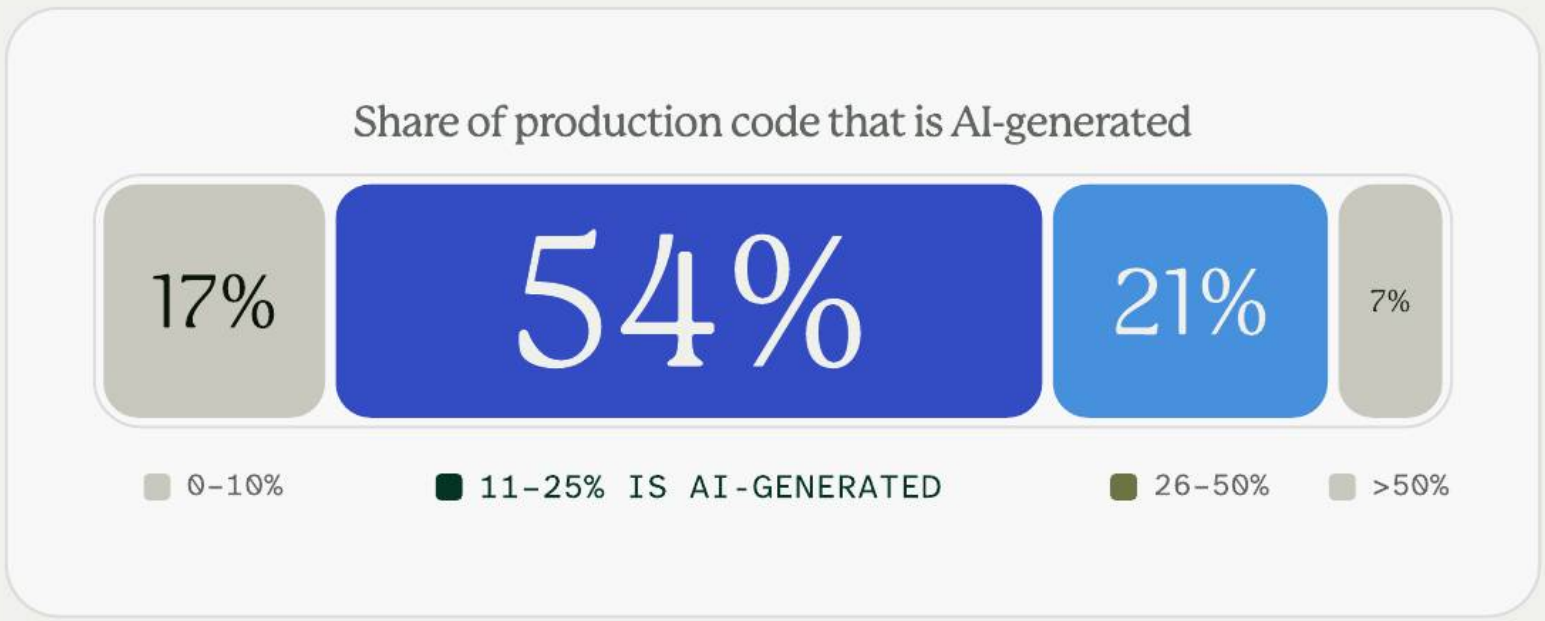
Who actually knows what's running in production?

THE AI-GENERATED CODE BASELINE

The AI code explosion has arrived

The shift is already well underway, with AI generating code faster than organizations can validate it. When asked what percentage of their production code over the past 12 months was written or substantially generated by AI tools, the respondents confirmed that AI-generated code is now a baseline fact of software development — not an experiment.

83% of organizations deploy production code that is more than 10% AI-generated.



More than half (54%) sit in the 11%–25% range, while 28% report that AI generates more than a quarter of their production code. A meaningful 7% are already past the halfway mark. Executives skew slightly less aggressive than engineering leaders on AI adoption (26% vs. 32% at the "more than 25%" threshold), suggesting the people closer to the work are pushing harder — or seeing more.

This is a meaningful shift from 2025, when the survey asked whether organizations were using agentic AI for testing at all, and 66% of those who were still described themselves as in a piloting phase. The 2026 question assumes AI-generated code as a given and asks how much. The pilots became the pipeline.

The correlation worth noting: Organizations that frequently or sometimes deploy with known testing issues are at 91% above the 10% AI-code threshold, compared to 73% for those who rarely or never do so. More AI-generated code correlates directly with more knowingly shipped defects. That's the code verification crisis in data form — and it runs through everything that follows in this report.

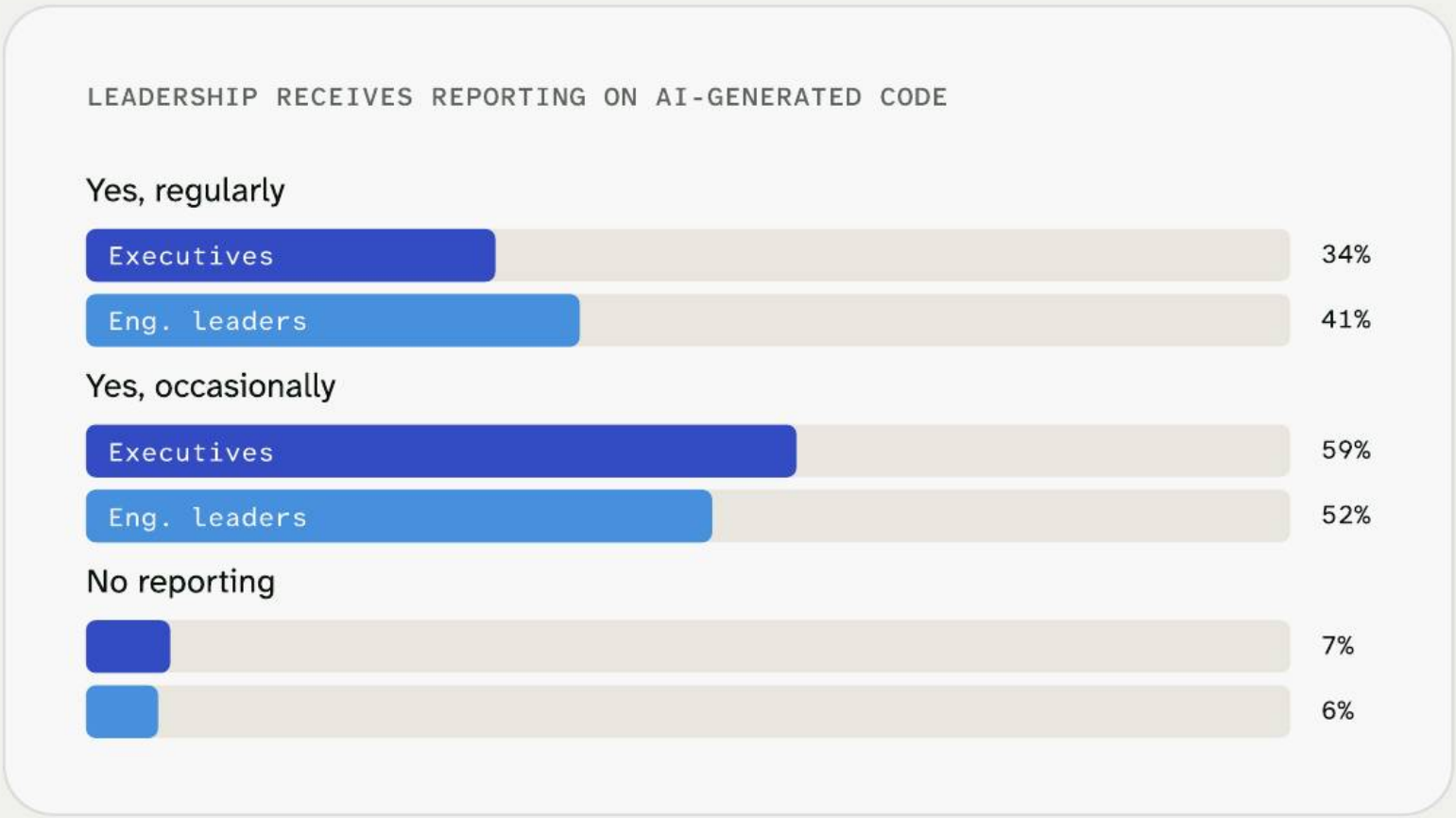
LEADERSHIP REPORTING

Visibility without accountability

Most organizations have at least some reporting on AI-generated code flowing to leadership. 93% of organizations with AI-generated production code say their leadership receives some form of reporting — 38% get it regularly, 55% occasionally.

But "reporting" and "understanding" are not the same thing. Only about one in three leaders gets a regular view into what their organizations are actually shipping. The majority are working from occasional updates, and nearly 7% have no visibility at all.

The numbers split in an interesting direction when you look at the exec/engineering divide: Engineering leaders of teams closest to the codebase are more likely to say their organizations report regularly (41% vs. 34%). Executives are more likely to receive only occasional reporting, meaning the people making strategic decisions may be working with the oldest information.



93% of organizations receive some form of leadership reporting on AI-generated production code — but only **38% get it regularly.** The people making strategic decisions may be working with the oldest information.

CUSTOMER DISCLOSURE

What are you actually telling users?

When organizations deploy software that contains AI-generated code, who knows? Customers and end users may not.

89% of organizations say they disclose to customers when software contains AI-generated code, at least sometimes. But only 27% disclose "consistently." The majority — 62% — do so only in certain situations.

Executives are more likely to say they disclose "in certain situations" (67%) than to say they do it consistently (23%). Engineering leaders skew the opposite direction, with 31% reporting consistent disclosure. Whether that reflects different levels of directness with customers, different internal policies, different understandings of the same policy, different assessments of regulatory exposure, or different incentives to be transparent is worth asking.

The 11% who say they do not disclose at all face increasing pressure as regulatory frameworks arrive. The EU AI Act is now in force, and "we disclose in certain situations" is not a compliance posture that most legal teams have formally reviewed against its requirements. The 62% in the "certain situations" bucket are making judgment calls about disclosure thresholds that belong in a policy document, not an engineer's discretion.

> Among organizations that have already traced a production incident to AI-generated code, 24% say they still don't disclose to customers, versus just 8% among those who haven't experienced such an incident. The organizations with the most direct evidence of AI-related risk are among the least transparent with customers about it.

SECTION 02

02

Resolving Testing Issues

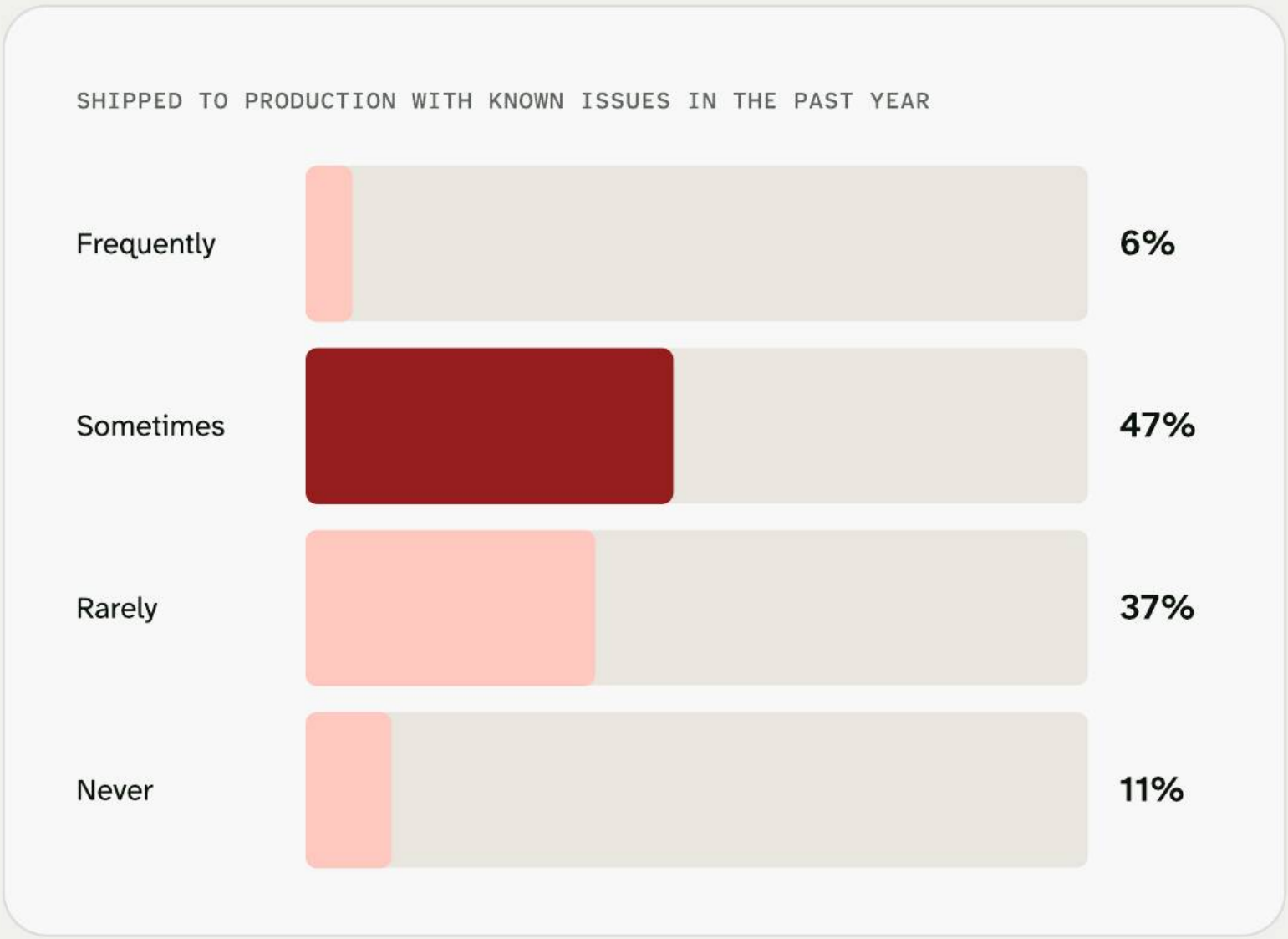
What gets shipped, and what it costs.

SHIPPING WITH KNOWN DEFECTS

More common than comfortable

"Testing complete" and "ready to ship" have become decoupled concepts

Deploying software with known or unresolved testing issues is a practice that most organizations would rather not discuss publicly. But the data suggests they're doing it anyway, whether due to leadership pressure, a competitive survival instinct, or a lack of resources for thorough testing.



53%

have shipped with known issues at least sometimes — identical for executives and engineering leaders.

Only 11% report they've never done it. The other 37% say it happens rarely — which, read generously, is good news, and read less generously, still means it happens.

THE COST OF THE MOST SIGNIFICANT INCIDENT

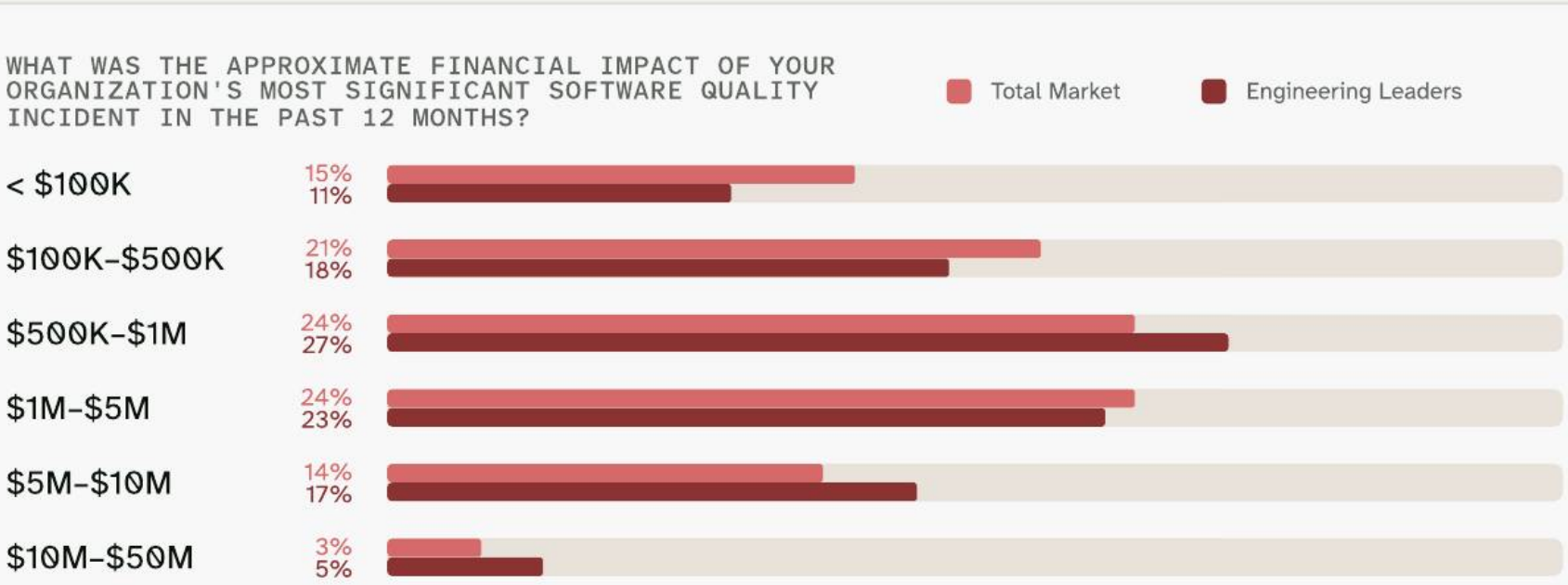
When quality fails, it isn't cheap

The data solidified in ways hard to ignore when respondents estimated the financial impact of their organization's most significant software quality incident in the past 12 months.

65% reported that their most significant quality incident cost $500,000 or more. More than four in 10 — 41% — said it hit $1 million or more. And 17% disclosed a worst-case incident costing $5 million or more.

Engineering leaders report higher-dollar incidents than executives across the board. At the $1 million-or-more threshold: 45% of engineering leaders versus 37% of executives. At the $5 million-or-more mark: 22% of engineering leaders versus 13% of executives.

There are two ways to read this: 1) Either engineering leaders have better visibility into the true cost of incidents — and are therefore reporting more accurately — or 2) they are dealing with more severe incidents than their executive counterparts realize. Neither interpretation is particularly reassuring.



One methodological note worth stating plainly: This question asked about the financial impact of each organization's most significant software quality incident generally, not whether that incident was caused by AI-generated code specifically. The data establishes that 80% of organizations have traced at least one production incident to AI code. Both facts belong in the same conversation, but attributing the dollar figures directly to AI causation would overstate what the data supports.

What the data does support: Organizations experiencing meaningful quality failures and organizations experiencing AI-sourced production incidents are, in large part, the same organizations.

INCIDENTS TRACED TO AI

The heart of the confidence gap

HAS YOUR ORGANIZATION TRACED ANY PRODUCTION INCIDENT, OUTAGE, OR CUSTOMER-IMPACTING DEFECT TO CODE GENERATED BY AI TOOLS IN THE PAST 12 MONTHS?

63%	17%	19%
Yes, once or twice	Yes, multiple times	No, never

Yikes! Among the organizations using AI-generated code, four out of five have already connected a real customer-facing incident to AI-generated code — 17% multiple times, 63% at least once or twice. Only 19% said they've never traced an incident to AI code, and 1% weren't sure.

Engineering leaders are slightly more likely to report these linkages (82% vs. 78% for executives) — again suggesting they have closer visibility into the actual failure modes.

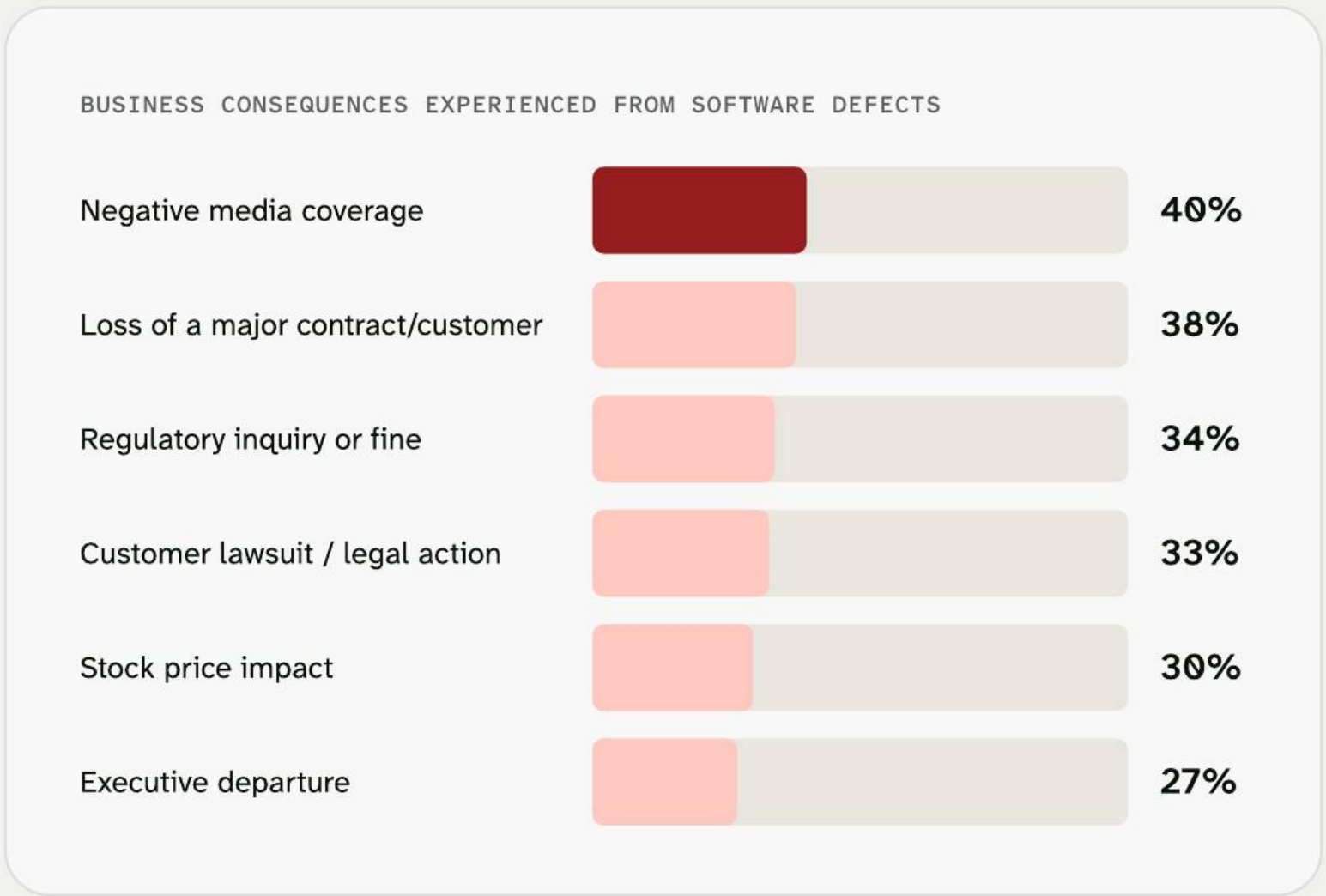
Organizations are scaling AI-generated code into production even as the majority are tracing incidents back to it. Unfortunately, these are not separate phenomena.

BUSINESS IMPACTS

Defects don’t stay technical for long

Software defects don't stay technical problems for long. We asked what business consequences organizations experienced from software defects in the past 12 months.

Nearly 90% reported at least one consequence.



95%

of organizations that traced an AI-code incident experienced real business consequences — vs. 54% of those that haven’t.

Engineering leaders consistently report higher rates of serious consequences than executives — most strikingly on executive departure (34% vs. 21%) and stock price impact (34% vs. 27%). The people closest to the work are more likely to say it cost the company a leader and moved the stock.

Among organizations that have traced a production incident to AI-generated code, 95% experienced real business consequences. Among those who haven't traced such an incident, that figure drops to 54%. The "not yet" in that 54% is doing a lot of work.

SECTION 03

03

The Workforce Evolution

What AI adoption is doing to the teams building software.

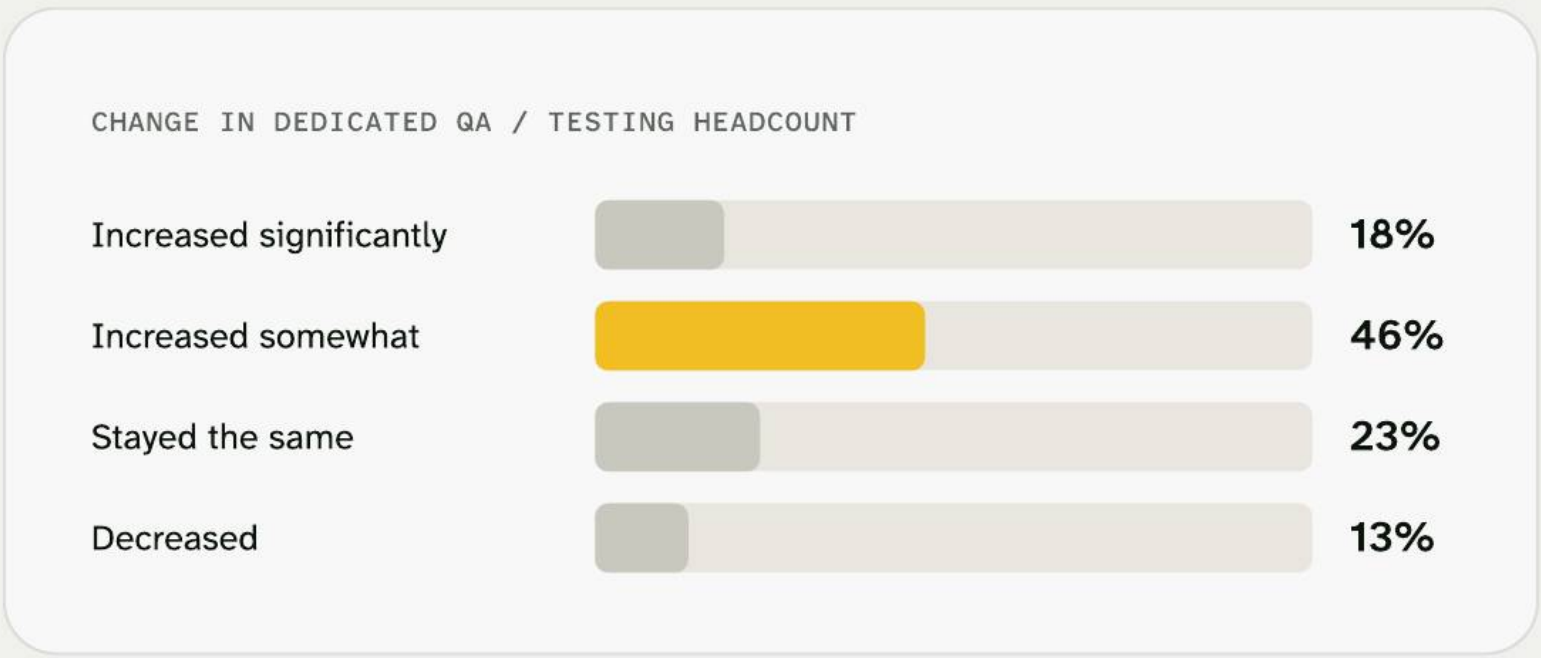
QA HEADCOUNT

Growing and shrinking at once

Does AI create more work for QA teams?

The story around AI and QA headcount turns out to be more complicated than the simple yet widely believed "AI will replace testers" narrative. Unfortunately, it's not overly reassuring in the opposite direction either.

64% of organizations report that their dedicated QA or testing headcount actually increased over the past year — 18% significantly, 46% somewhat. Only 13% saw a decrease.



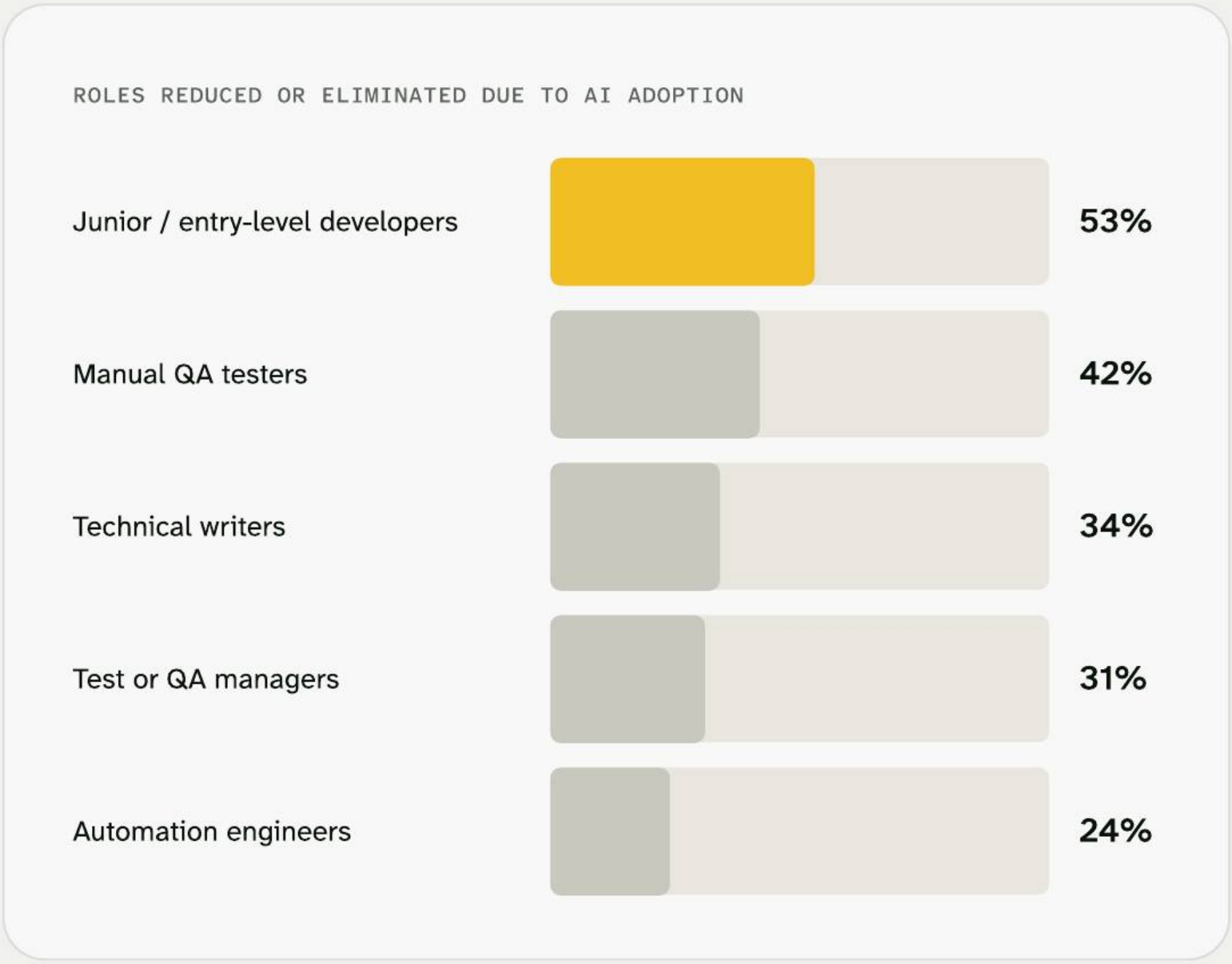
At first glance, that looks like a vote of confidence in human testers, but the mechanism here matters. Among organizations that have traced production incidents to AI-generated code, 67% increased QA headcount — compared to 51% of those who haven't traced such incidents. AI's not replacing QA teams but generating enough additional code volume and quality risk that organizations are hiring more QA to keep up. The code verification crisis again: more AI, more throughput, more work to verify.

In 2025, the workforce impact was still largely hypothetical: 54% of respondents said they'd be comfortable with AI agents determining job cuts, with engineering leaders (61%) significantly more comfortable than executives (47%). In 2026, the actual shape of that shift is documented. The picture shifts when you look at which roles are being eliminated — and the entry-level hiring data alongside it. Adding senior headcount while cutting junior and specialized roles is a consolidation, not a renaissance.

ROLE ELIMINATIONS

Where the cuts are landing

Nearly nine in 10 (84%) organizations have eliminated or significantly reduced at least one role due to AI adoption. The breakdown reveals that roles with the most procedural overlap with AI capabilities are going first.



53%

have eliminated or reduced junior / entry-level developer roles due to AI adoption.

The concentration of cuts at the junior and entry level is particularly significant, revealing a talent pipeline story. The people who historically learned the craft of software development by handling the procedural, error-prone work that AI now absorbs are no longer getting those entry-level jobs. Tomorrow's senior QA professionals are diminishing as AI generates more code, creating corresponding quality risks. And while the human oversight needed to manage these challenges has become an even larger responsibility than before, the ability to build the necessary future expertise is being compromised by the lack of foundational experience in the workforce.

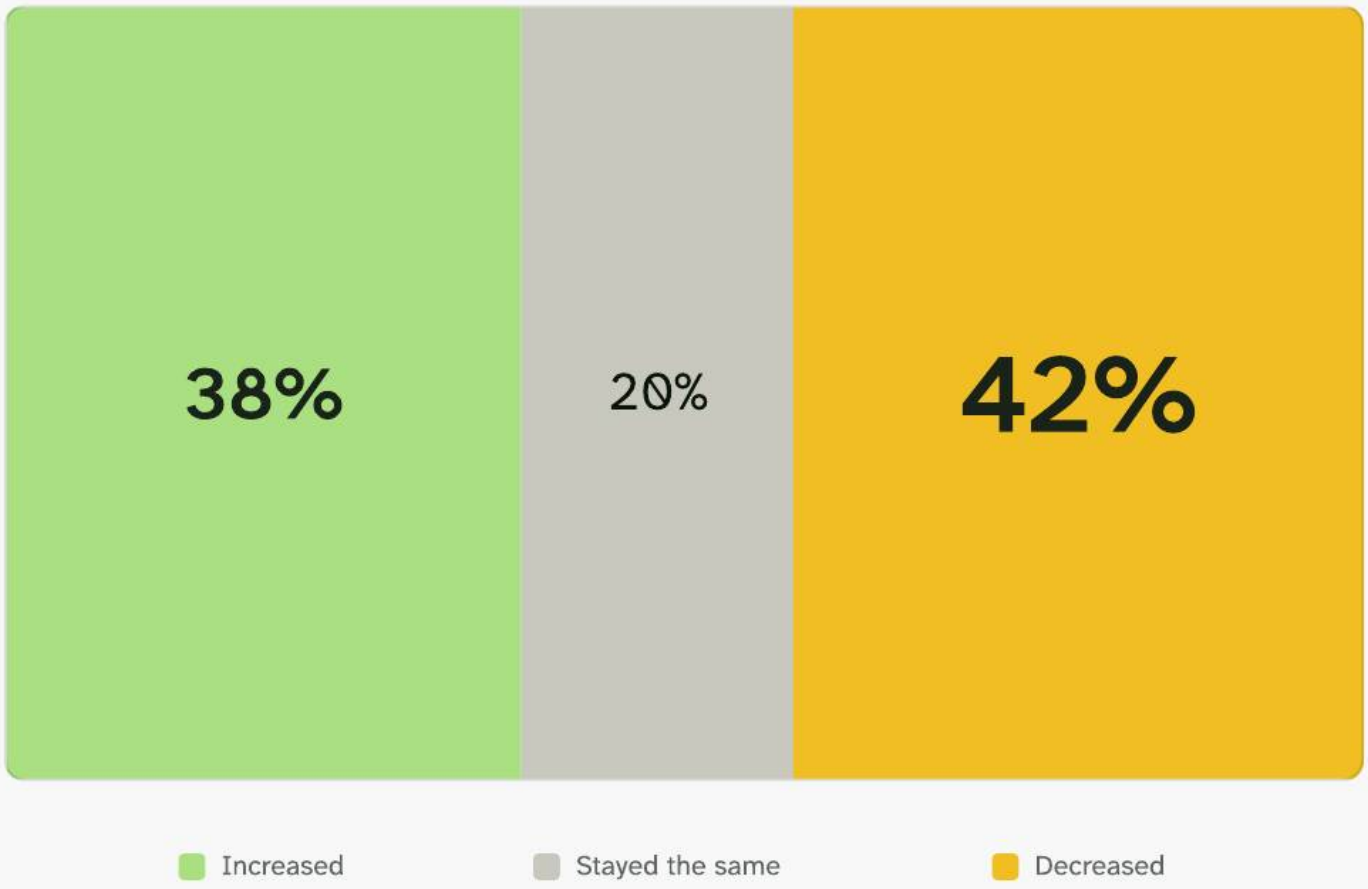
ENTRY-LEVEL HIRING

A pipeline under pressure

Entry-level hiring data confirms the trajectory.

More than four in 10 (42%) organizations say entry-level software engineering hiring has decreased compared to the prior year — 32% somewhat, 11% significantly. Only 38% increased entry-level hiring.

CHANGE IN ENTRY-LEVEL ENGINEERING HIRING VS. PRIOR YEAR



What's clear: The pathway into software development as a profession is narrowing, at exactly the moment that the complexity of overseeing AI-generated code is intensifying.

Engineering leaders are more likely to report increasing entry-level hiring (42%) than executives (33%), though executives are more likely to say hiring has decreased significantly (14% vs. 8%). The gap may reflect different vantage points on the same decisions or different levels of information.

SECTION 04

04

The Trust-Risk Imbalance

What leaders worry about vs. what the data shows they should.

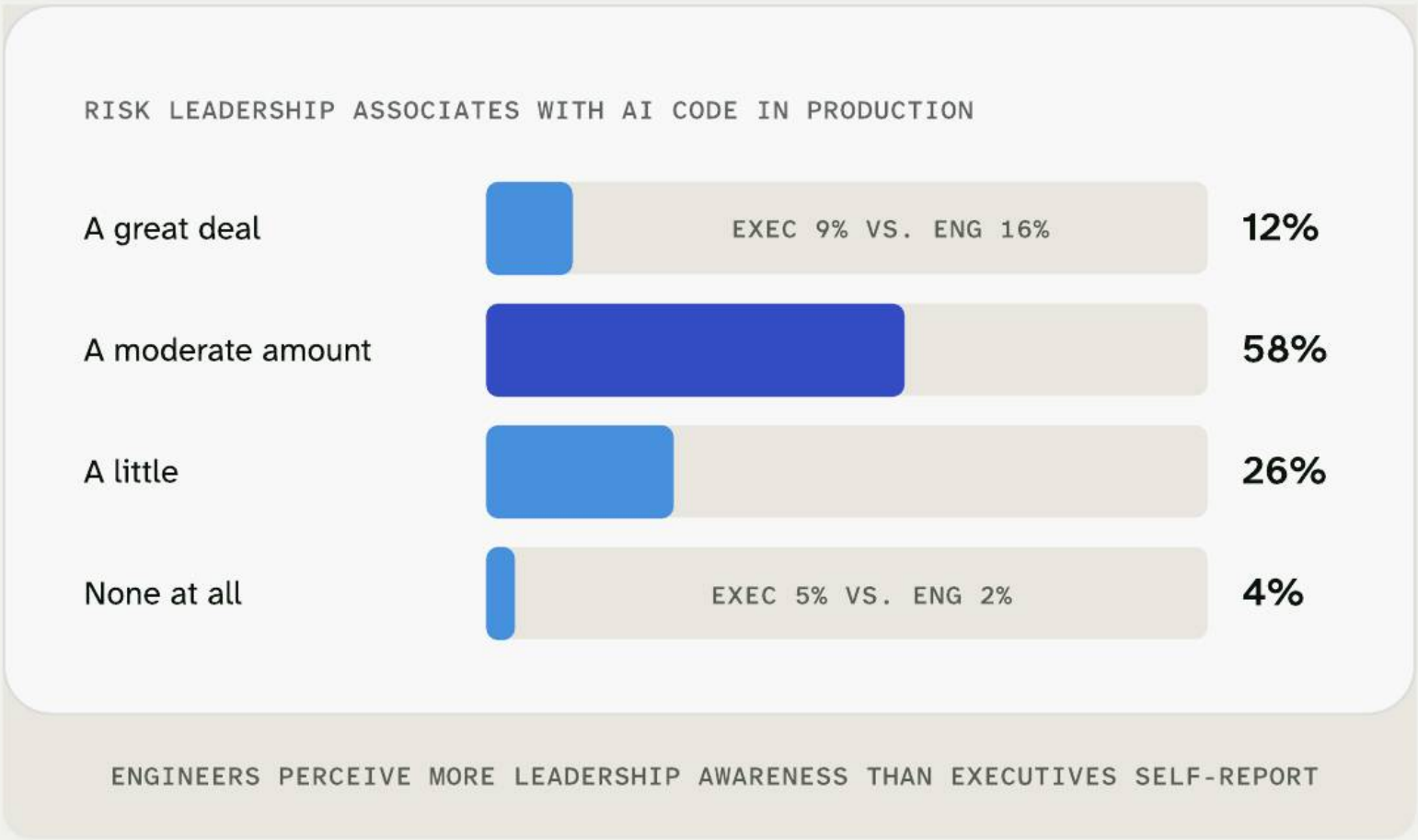
LEADERSHIP RISK PERCEPTION

Moderate, not alarmed

Risk awareness exists, but does the urgency to act on it?

When asked how much risk organizational leadership associates with AI-generated or AI-tested code reaching production, the dominant response was "a moderate amount" — chosen by 58% of respondents.

Seven in 10 (70%) said their leadership perceives "a great deal" or "a moderate amount" of risk. Only 4% said their leadership perceives none.



Even though that sounds like healthy risk awareness, "a moderate amount" is doing a lot of work in a world where **80% of organizations have traced production incidents to AI-generated code and 40% have absorbed $1-plus million quality failures.**

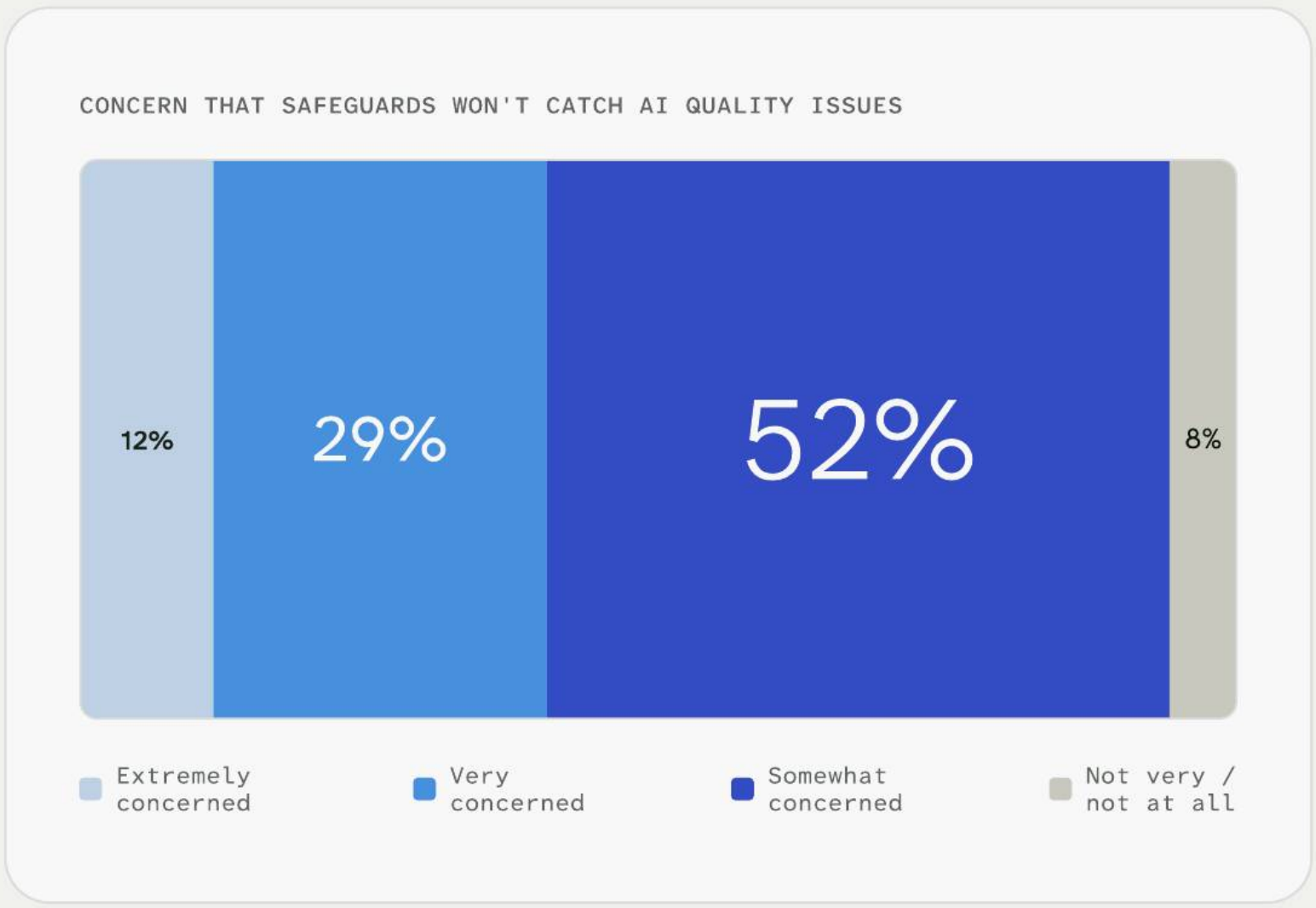
Engineering leaders are notably more likely to perceive "a great deal" of risk (16%) than executives (9%). The people with the most direct exposure to the failure modes are most likely to flag the severity, whereas those making strategic bets are more likely to rate it "moderate."

SAFEGUARD CONFIDENCE

Concerned, broadly

The unease becomes explicit when respondents detailed concerns that their organization's current safeguards are not effective at preventing AI-generated or AI-tested features with quality or safety concerns from reaching production.

More than nine in 10 (92%) expressed some level of concern. Among those: 12% are extremely concerned, 29% are very concerned, and 52% are somewhat concerned. Just 8% said they're not very or not at all concerned.



92%

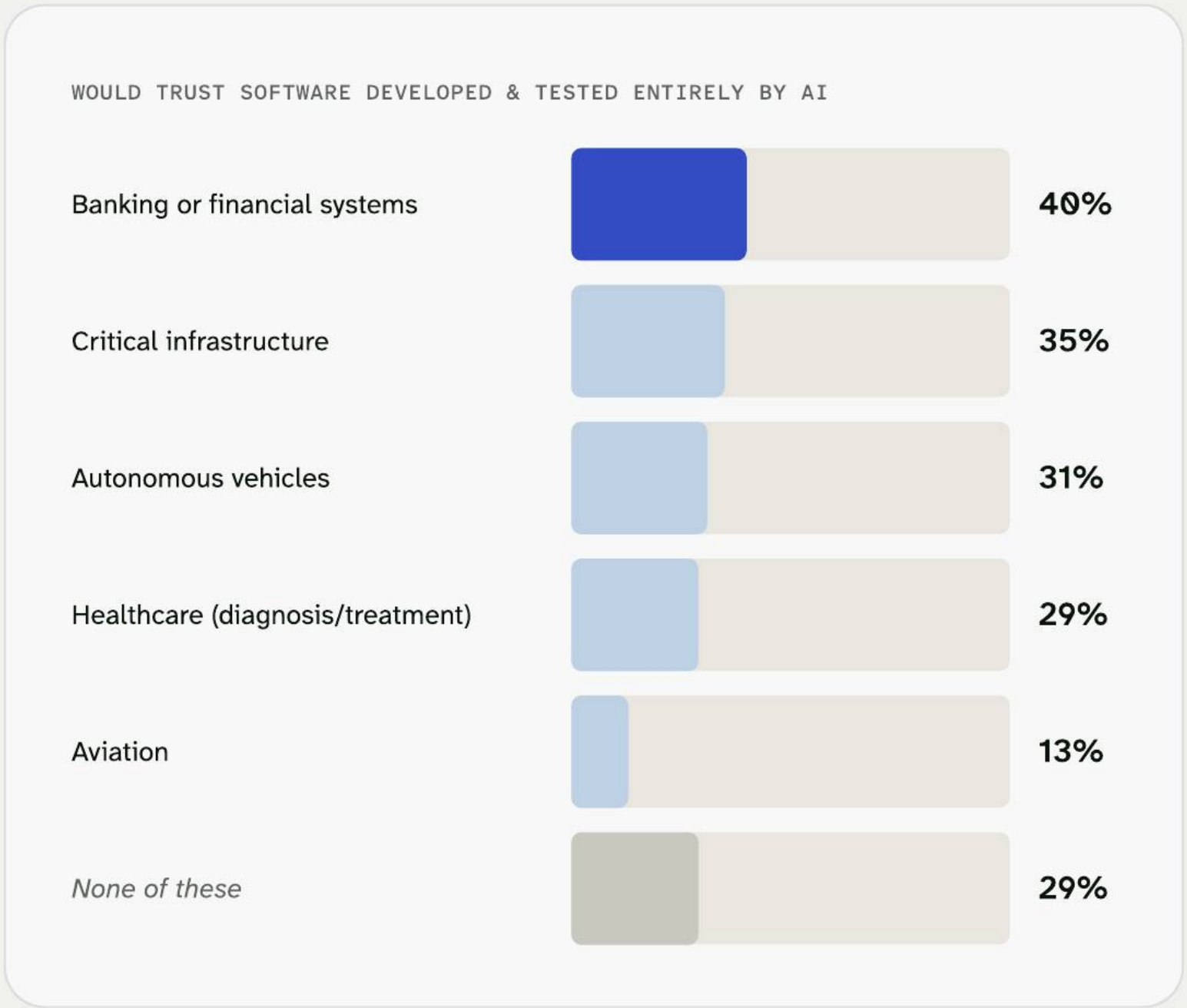
are at least somewhat concerned their safeguards won't catch AI-generated quality issues before production.

The "extremely or very concerned" number is 41%, split evenly between executives and engineering leaders — a shared alarm. Nearly everyone is worried. Most of them are doing something about it or intend to.

TRUST BY DOMAIN

Where AI-only is a hard no

The survey revealed where people would personally trust software developed and tested entirely by AI, without any human involvement. The results show significant variation across different domains, along with a substantial minority who would trust AI-only software in none at all.



Aviation sits at the bottom of every list, in every segment, with only 13% trusting AI-only software in that sector. Nearly 30% would trust it in nothing, full stop.

The findings suggest that even among respondents who are actively deploying AI-generated code and testing tools, full autonomy in high-stakes domains remains a bridge too far. The question is whether the organizational incentives to move fast are outpacing the instincts that say slow down.

SECTION 05

05

The Pressure Cooker

Why quality gets traded for speed —
and who knows it's happening.

QUALITY COMPROMISED FOR DEADLINES

An open secret

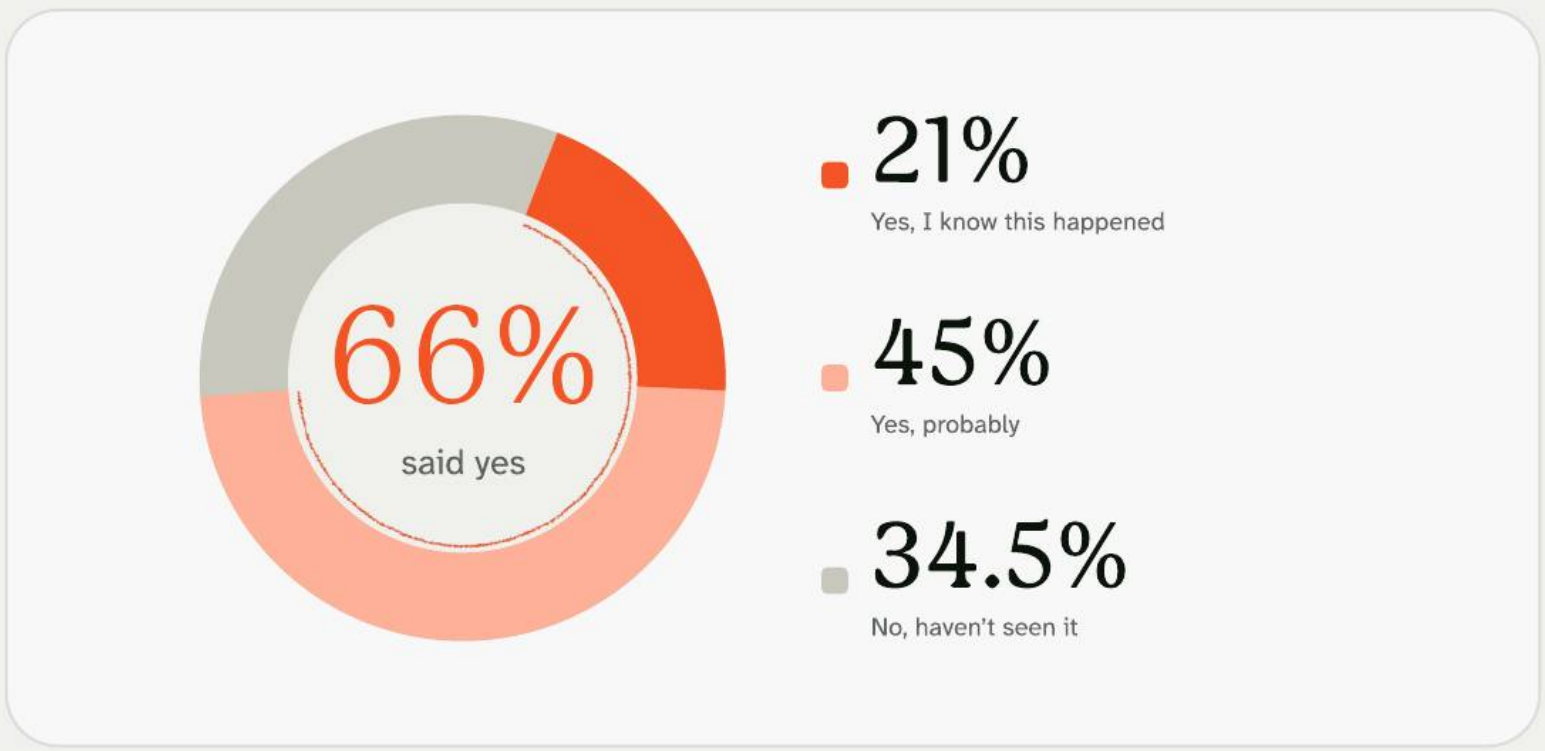
Software quality is being traded for speed

The survey asked whether teams at their organizations had compromised on quality or testing standards to meet release deadlines in the past 12 months.

66% said yes — either definitely (21%) or probably (45%).

The people saying "yes, I know this has happened" are split evenly between the two groups: 21% of executives and 21% of engineering leaders. Where they differ is on the "probably": executives are more likely to acknowledge it with uncertainty (49% vs. 41%), suggesting that the closer you are to the work, the clearer the picture becomes.

Only 34.5% of respondents said they had not seen this happen.



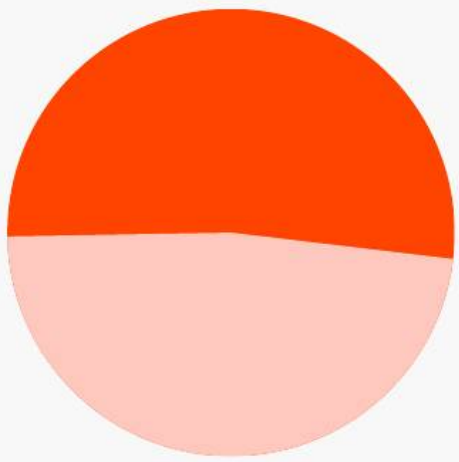
Sit with that for a moment. Engineering teams are being directed to adopt AI to generate code faster. The volume grows, but the testing infrastructure wasn't built to handle it. Quality gets traded for speed because the math of what teams are asked to do no longer adds up. The majority of organizations are knowingly or probably compromising on quality to hit release dates, with 53% shipping with known, unresolved issues at least some of the time. Compromising standards and shipping anyway are two sides of the same pressure. Against the backdrop of a world where 80% have traced production incidents to AI-generated code, these findings translate to a meaningful operational risk.

THE AI DEPLOYMENT DILEMMA

Speed vs. safety

Respondents indicated which risk concerns them more for the next 12 months: deploying AI too quickly or falling behind by not deploying it fast enough.

The split was almost precisely even. **53% said deploying too quickly is the greater risk compared to the 47% who said falling behind is.** Executives and engineering leaders gave identical answers.



53%

Deploying AI too quickly is the greater risk

47%

Falling behind by not deploying fast enough is the greater risk

Speed vs. safety is the clearest expression of the bind organizations find themselves in, with no consensus path. Teams that move fast risk the quality failures the data shows are already happening. Teams that move cautiously risk competitive irrelevance. The 53/47 split illustrates the standoff. Orgs are caught in a cycle where the mandate to adopt AI and the infrastructure to govern it are moving at different speeds, and no one has formally decided which matters more.

SECTION 06

06

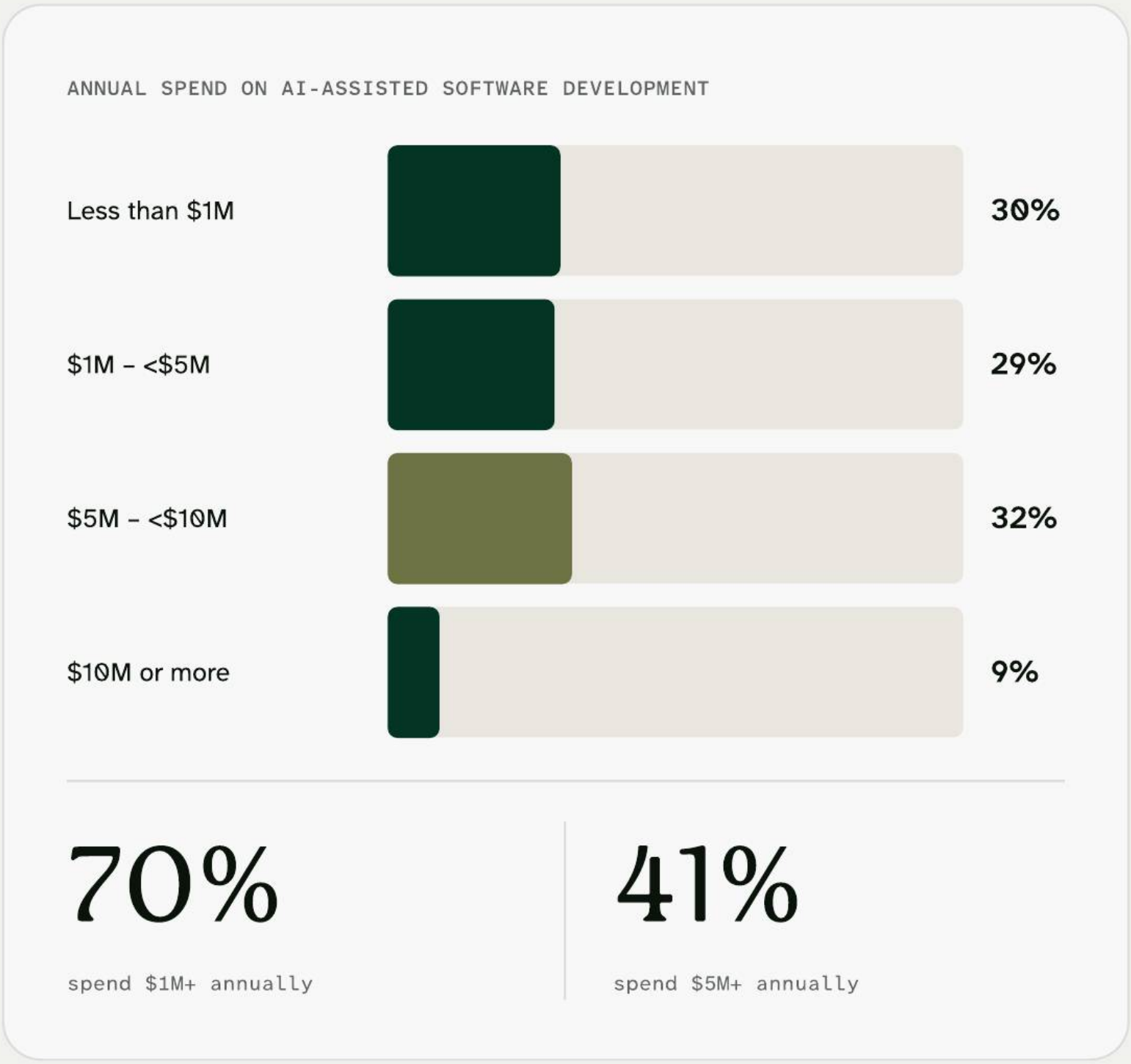
Spending & ROI

What organizations put in, and what they say they get back.

THE INVESTMENT PICTURE

Substantial, and concentrated

Spending on AI-assisted software development is substantial and concentrated at the upper end of the various ranges. **Seven in 10 organizations spend $1 million or more annually** on AI-assisted software development, with **41% spending $5 million or more.**



The most common single bracket: **$5 million to less than $10 million (32%).** Executive and engineering leader spending patterns are nearly identical, suggesting budget allocation decisions are being made with similar information on both sides of the organizational chart.

RETURN ON INVESTMENT

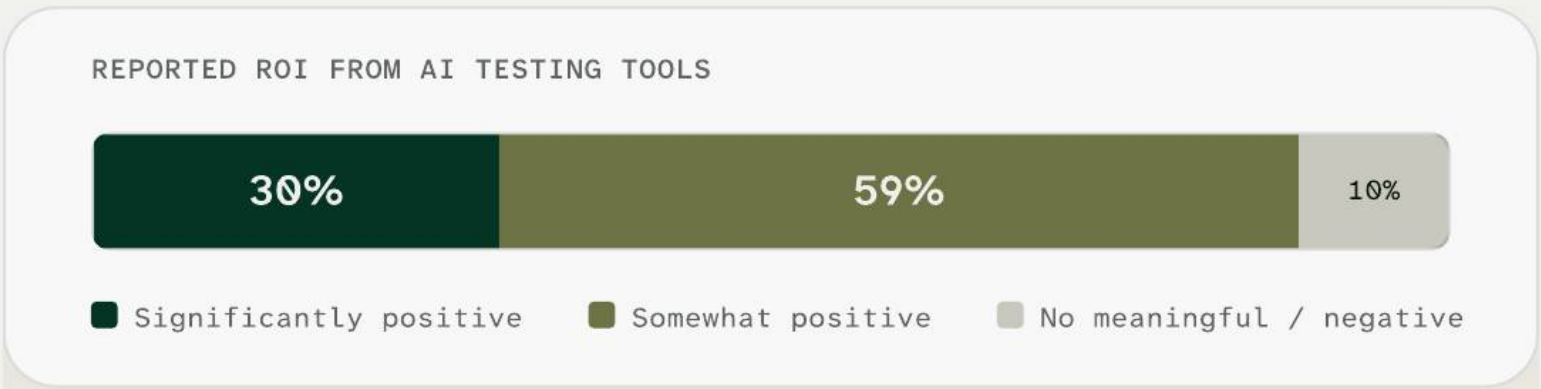
Positive but uneven

Given the scale of investment, how do leaders characterize their returns from AI testing tools specifically?

Nearly nine in 10 (89%) report positive ROI — 30% significantly positive and 59% somewhat positive, while only 10% report no meaningful ROI or negative returns.

Executives are more likely to report significantly positive ROI (34%) than engineering leaders (26%), suggesting that the framing of "what counts as ROI" may differ between the two groups, or that executives are applying a higher-level lens that registers gains more readily.

The 10% reporting negative or no meaningful ROI exhibit early signals from organizations that may be in the most honest position to report whether the investment math actually works.



The more difficult question embedded in the 89% positive figure: What is that ROI actually measuring? The data shows that the same organizations reporting positive ROI are, in large numbers, also tracing production incidents to AI-generated code (80%), expressing concern about their safeguards (92%), and shipping with known defects (53%). That combination suggests ROI is being measured on output — shipping faster, generating more code — rather than on quality outcomes: fewer incidents, lower defect costs, reduced blast radius when something goes wrong. Speed is legible, yet quality debt accumulates quietly.

This finding also marks a notable shift from 2025, when only 55% of organizations reported expecting financial benefit from AI testing tools. The jump to 89% positive ROI in 2026 reflects real adoption gains, but the incident and safeguard data suggest the full cost of that speed hasn't yet landed in the accounting.

At a moment when the broader AI investment narrative has grown skeptical — with mounting questions about whether AI tooling budgets are generating measurable returns — the software testing finding stands out.

SECTION 07

07

What’s on the Horizon

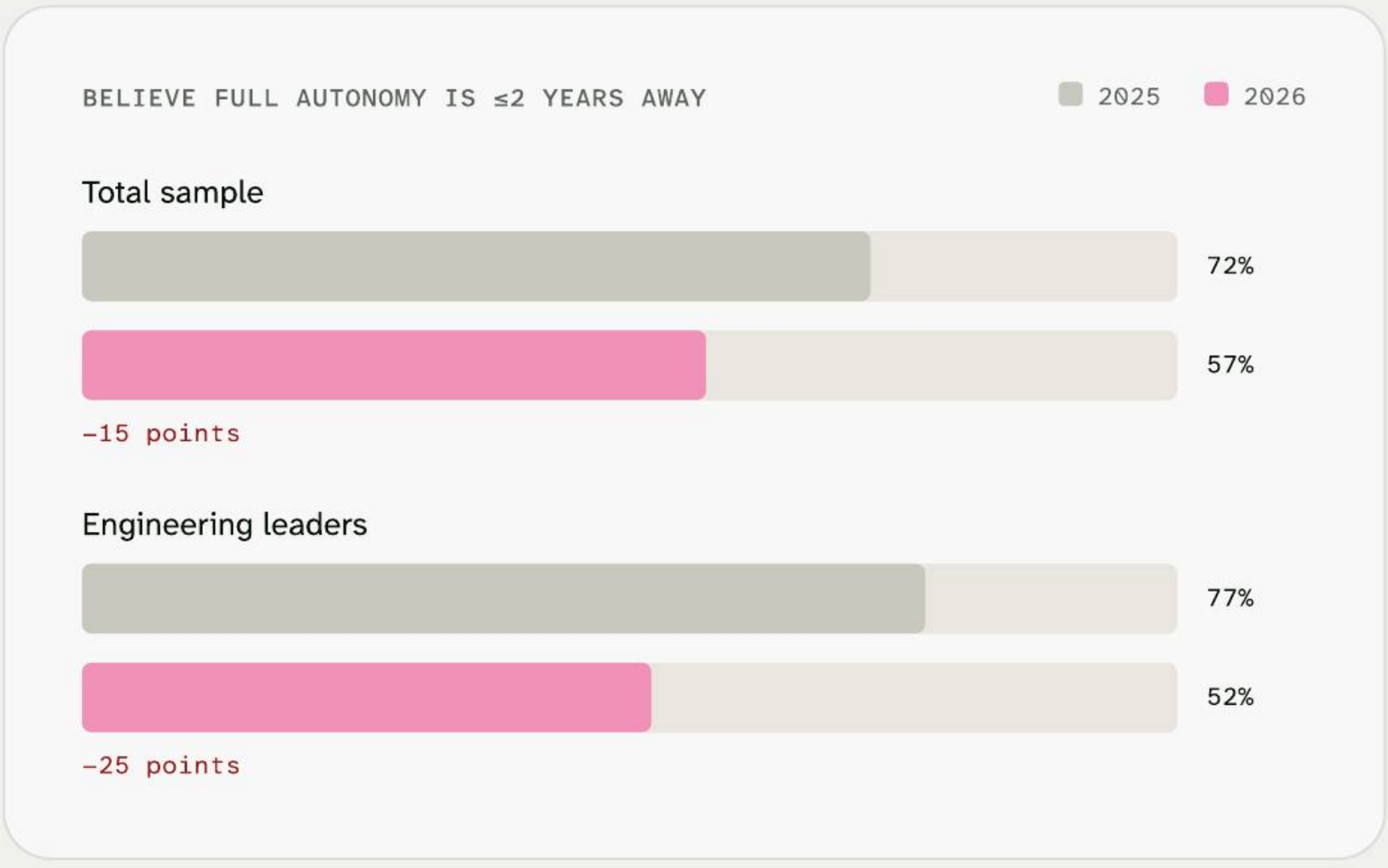
What the future holds, and who’s ready for it.

THE AUTONOMOUS TESTING TIMELINE

Expectations are being revised down

One of the most closely watched questions in enterprise AI adoption concerns full autonomy: When, if ever, will AI be reliable enough to test and deploy software without human involvement?

The largest single group — 27% — believes fully autonomous testing and software deployment will happen within two years. Another 24% say within three years. But 8% say it's already possible today.



Executives are more optimistic: 62% expect full autonomy within two years, compared with 52% among engineering leaders. Engineering leaders are more likely to say "never" (3% vs. 0%), an option that didn't exist in last year's survey. The people who've seen the failures up close are more cautious about the timeline.

The year-over-year shift is significant: In 2025, 72% of respondents believed that fully autonomous testing would be achievable within 2 years. However, this figure has since declined to 57%, marking a 15-point decrease over the past year. Among engineering leaders, the shift was even more pronounced, dropping from 77% to 52%, a 25-point decline. The people closest to the tools are revising their expectations the fastest.

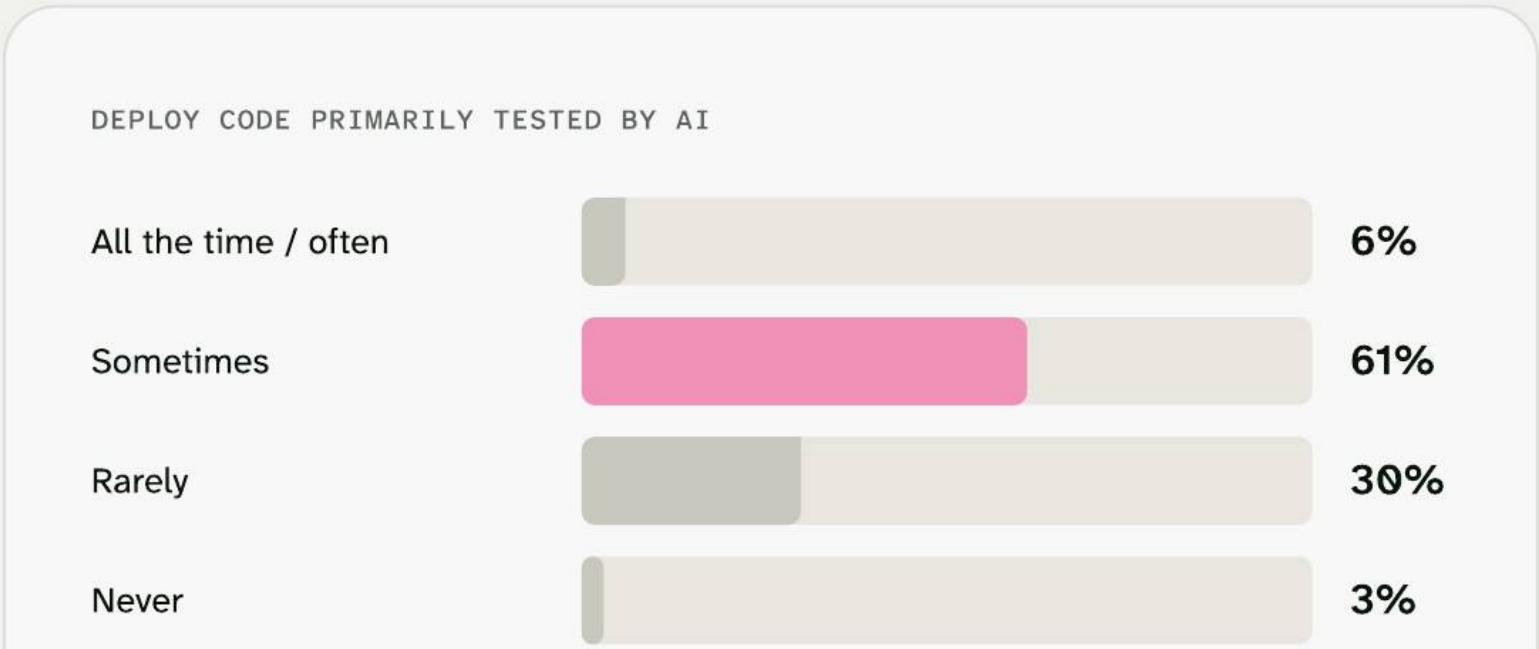
AI-TESTED CODE IN PRODUCTION TODAY

The future is already shipping

While the debate about full autonomy continues, organizations are already deploying AI-tested code into production — just not at full scale yet.

More than six in 10 (66%) organizations deploy code that has been primarily tested by AI at least sometimes. Only 3% say never. The "sometimes" bucket is dominant at 61%, with 6% deploying AI-tested code often or all the time.

A critical nuance: The question about "full autonomous testing arriving someday" misses the more immediate reality that organizations are already deploying AI-tested code at a meaningful scale today. The infrastructure for full autonomy is not waiting for some future green light.



66% of organizations already deploy code that has been primarily tested by AI at least sometimes.

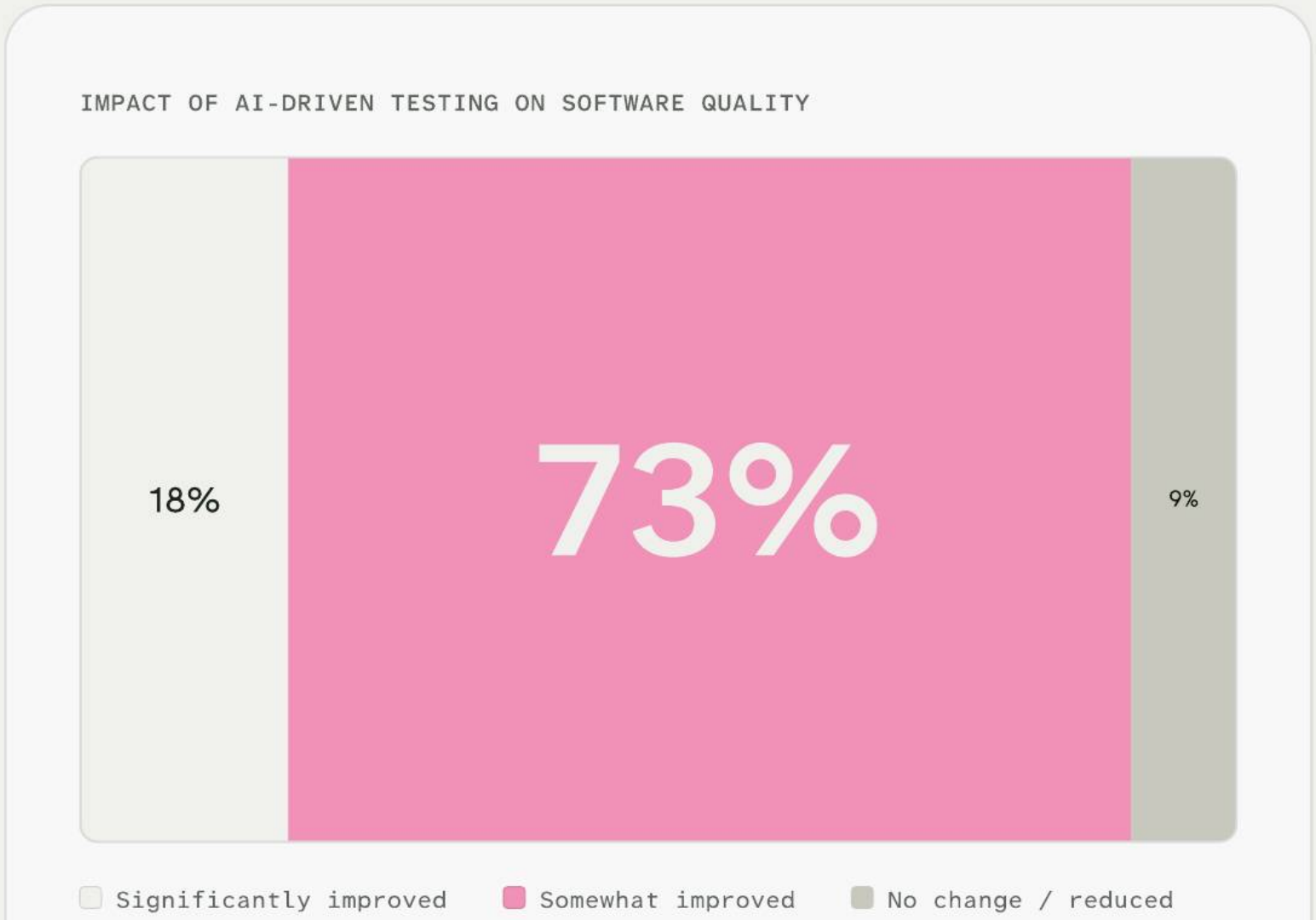
In 2025, 85% of respondents said the ideal testing approach was a hybrid of human and AI, with 44% preferring mostly AI with some human oversight, 40% preferring mostly human with some AI. Only 12% wanted AI-only testing. The 2026 behavioral data confirms that hybrid is exactly what's happening in practice: 66% deploy AI-primarily tested code at least sometimes, but only 6% do so all the time or often. Human review remains the default final gate, even as AI handles more of the work upstream.

AI'S IMPACT ON SOFTWARE QUALITY

Better, but not without reservations

When respondents were asked to characterize the overall impact of AI-driven testing on software quality within their organizations, responses were broadly positive but not uniformly so.

91% say AI-driven testing has improved software quality — 18% significantly, 73% somewhat. Only 3% say quality has been reduced.



Engineering leaders are more likely to report significant improvement (21%) than executives (14%). They're also less likely to report quality reductions (1% vs. 5%).

The positive overall picture sits alongside the earlier data points — 80% tracing production incidents to AI-generated code, 53% shipping with known defects, 66% compromising quality for deadlines — without contradicting them. "Better than it was" and "not yet where it needs to be" can both be true at the same time.

SECTION 08

Synthesis

Closing the Confidence Gap

What the data tells us about what to do next

The findings in this report tell the story of an industry amid transition: more AI-generated code, more investment, generally positive returns on that investment, and broadly shared optimism about where the technology is heading.

The code verification crisis is real, and the confidence gap is expanding. The organizations that close it first will build the infrastructure that makes speed trustworthy. Here's what the data suggests that requires.

01

Stop treating reporting as accountability.

93% of organizations say leadership receives updates on AI-generated code in production, yet 80% have still traced a production incident to that code. Having a dashboard is not the same as having control. The gap between visibility and prevention is where the quality crisis lives. Close it by connecting what leadership sees to what actually determines a release decision — not as a retrospective artifact but as a precondition for shipping.

02

Name what's actually driving your AI adoption.

The data shows a 53/47 split on whether deploying AI too fast or falling behind poses the greater risk. That near-perfect standoff suggests most organizations haven't formally answered the question. When 66% are compromising on quality to meet deadlines and 45% aren't certain it's even happening, the strategy is not deliberate. But the more uncomfortable implication is directional: If the mandate to adopt AI is coming from the top, and the quality consequences are landing at the bottom, the people generating the failures are not the ones who designed the conditions for them. Articulating the actual driver — competitive pressure, board mandate, measured productivity gains — is the first step toward building a quality posture that matches it.

03

Redefine what ROI from AI testing measures.

89% report positive ROI from AI testing tools. The organizations reporting positive ROI are, in large numbers, also the ones tracing production incidents to AI-generated code and expressing concern about the adequacy of safeguards. Speed and output volume are legible metrics. Incident rate, defect escape rate, and mean time to detection are harder to measure, but they're what the investment is actually for. Organizations that add quality outcomes to their AI ROI definition will make better decisions about where to spend and where to slow down.

04

Treat the entry-level pipeline as a strategic asset.

53% of organizations have reduced or eliminated junior developer roles due to AI adoption, and 42% have decreased entry-level engineering hiring this year. While those decisions make short-term sense, they may not make long-term sense for organizations that will need a generation of engineers who understand how to oversee, interrogate, and govern AI-generated code at scale. Redefining junior roles — rather than eliminating them — is an investment in the institutional knowledge that can catch what AI misses.

05

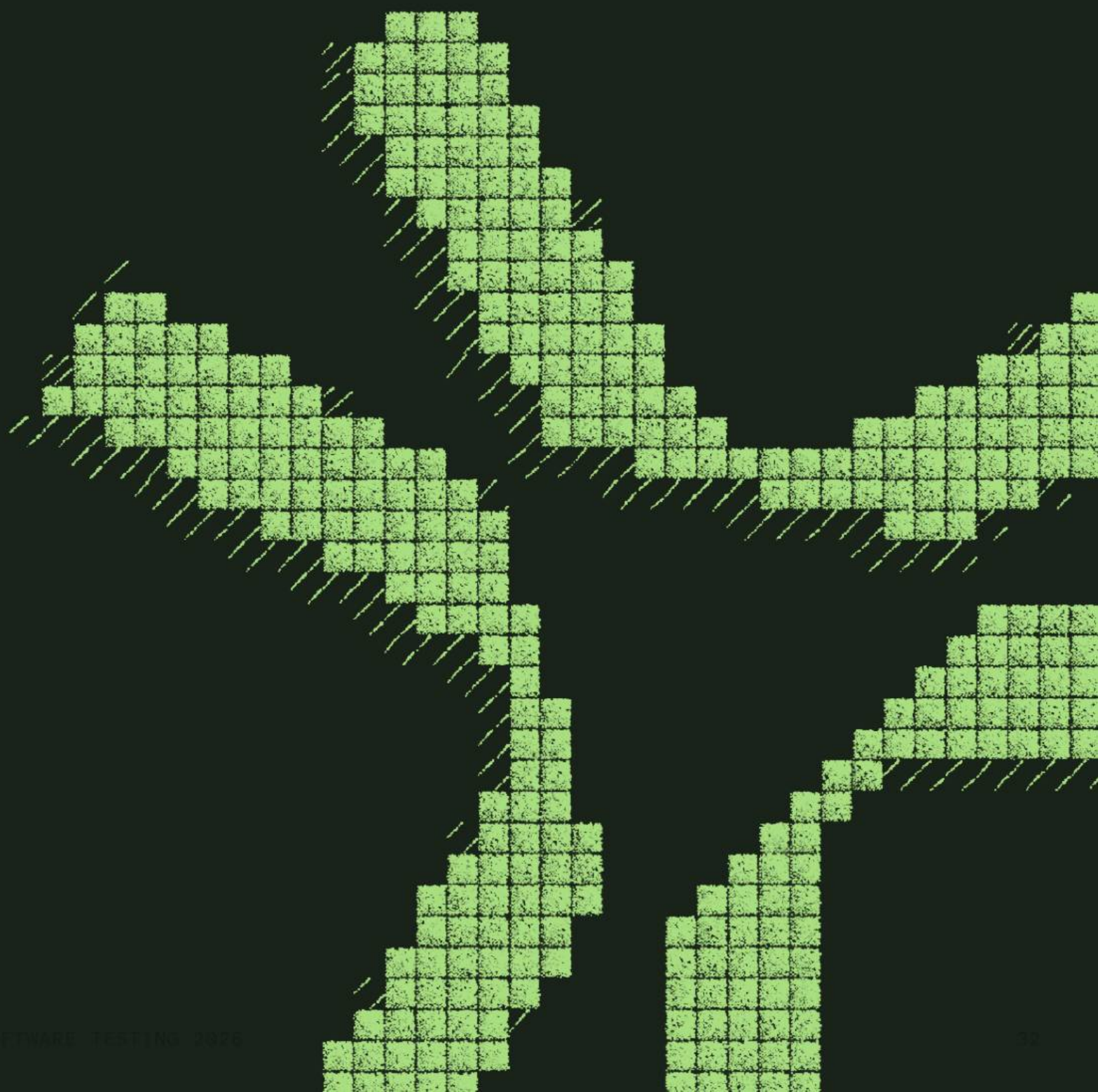
Build governance before the regulator asks.

Only 27% of organizations consistently disclose to customers when software contains AI-generated code. 34% have already faced a regulatory inquiry or a fine due to a software defect. With the EU AI Act in force, the organizations that treat disclosure as a legal liability to manage are one step behind those that treat it as trust to build. Clear, auditable records of what was known about AI-generated code, when it was known, and what was done about it serve as the evidentiary foundation for every regulatory, legal, and customer relationship conversation that comes next.

Is this code actually ready to ship?

This report keeps circling back to that question, and the data says most organizations still can't answer it confidently. Sauce Labs built AURA — the AI-unified release assurance platform — to close that gap, giving engineering teams and executives a shared, verifiable answer instead of disparate dashboards and hope that the code works.

AURA brings together the signals that live across your pipeline — test results, coverage gaps, error patterns, known issues — into a unified view of release readiness that doesn't require manual aggregation, informal status checks, or a leap of faith before you deploy.



Your Business is Unique. Your Testing Problems are Not.

Sauce Labs built the category once already. The company behind Selenium and Appium created the most widely adopted test automation platform of the cloud and mobile eras. AURA is that same team redefining the category again, this time for what enterprise testing needs to look like in the AI era: release assurance.

That redefinition rests on a data and expertise advantage most vendors can't match: a test execution dataset of more than 8.7 billion runs — the largest in the industry — built over 20 years of working inside the release pipelines of 80%+ of the world's top 10 financial institutions, a quarter of the Fortune 2000's IT software and services firms, and 60% of the top 10 global gaming developers. More than 300,000 enterprise users run on the platform today.

AURA:
AI-Unified Release Assurance

AURA is an agentic AI platform that closes the loop from business intent to production confidence — authoring test cases, running them, and analyzing the results at AI speed, with a human reviewing and approving every step.

Human-in-the-loop design directly answers a gap this report surfaces elsewhere: Only 27% of organizations consistently disclose to customers when software contains AI-generated code, and 34% have already faced a regulatory inquiry or a fine tied to a software defect. AURA keeps a documented, auditable trail of what was tested, what passed, and what didn't, built as the release progresses rather than reconstructed after an incident. Governance stops being a retrospective exercise and becomes a byproduct of how the release got shipped.

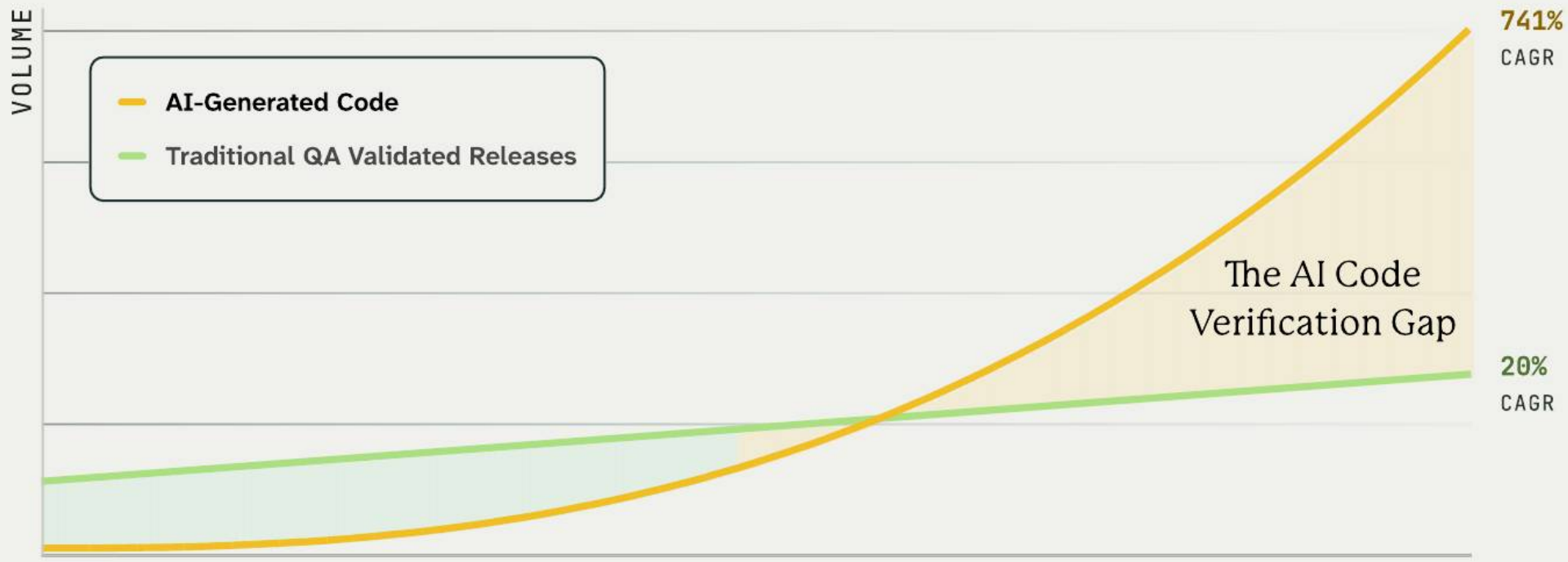
Real-World Impact

Across Sauce Labs' customer base — independently validated by Forrester's Total Economic Impact™ study and corroborated by Enterprise Strategy Group research — closing the verification gap results in fewer incidents, faster releases, and reclaimed engineering time. Here's what customers report on average:

Avg. 90%	Avg. 47%	Avg. 38%
REDUCTION IN PRODUCTION INCIDENTS	**FASTER RELEASE CYCLES TO MARKET**	**RECLAIMED ENGINEERING CAPACITY**
75% decrease in crashes	**90%+** automated test coverage	**80%+** faster test authoring
4x reduction in critical issues	**75%+** decrease in execution time	**~100%** device-agnostic tests
75% reduction in error resolution costs	**90%+** faster root-cause analysis	automated testing on **10,000+** devices

The AI-generated code verification crisis is here

AI-generated code is growing **exponentially** while test coverage is growing **linearly**



For this year's report, the ambition hasn't faded, and the spending hasn't slowed. However, the confidence gap has opened into something more difficult to merely paper over: an emerging code quality verification crisis that sits right at the intersection of a technology organizations are betting on and the ensuing challenges they haven't yet figured out.

How AURA Works

The AI-Unified Release Assurance Platform

[01] Author

Transforms plain-language intent into working test cases, so writing and maintaining test scripts stops being a full-time job for someone on the engineering team.

[02] Test

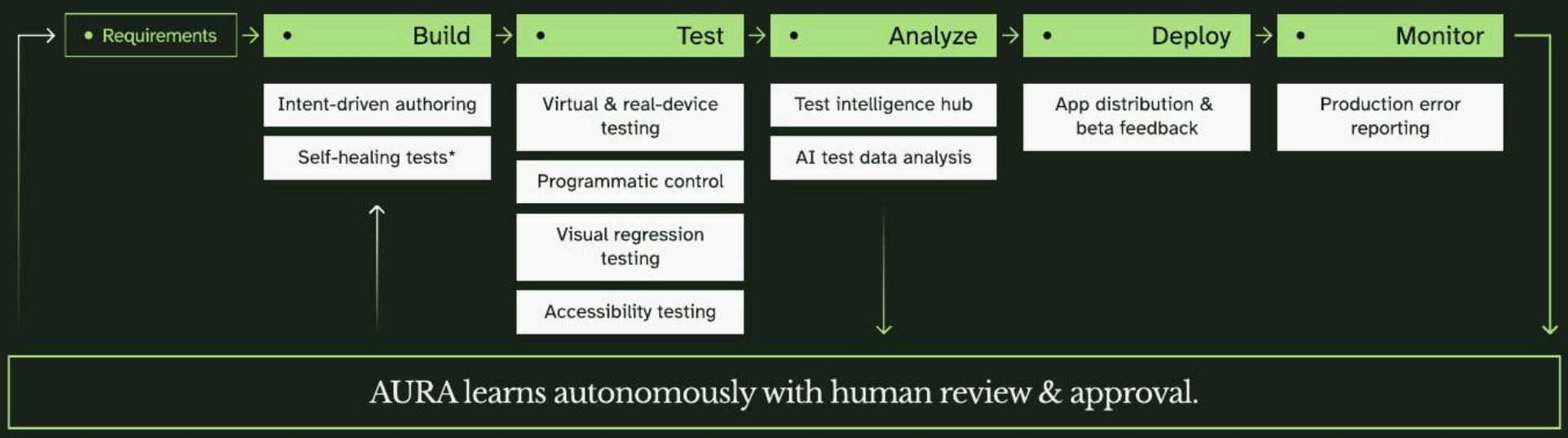
Runs those cases automatically across Sauce Labs' hosted device clouds, framework-agnostic and integrated directly into existing CI/CD pipelines.

[03] Analyze

Turns a test failure into a root cause rather than a time-consuming investigation, minimizing the time between something breaking and knowing why.

Full-Lifecycle Release Assurance

Not just bolted onto one part of the release pipeline, AURA runs the full lifecycle, from requirements through monitoring, with a measurable outcome at each stage:



BUILD	80%+ Faster	test creation, through intent-driven authoring and self-healing tests
TEST	47%+ Faster	release cycles, through virtual and real-device testing, programmatic control, visual regression, and accessibility testing
ANALYZE	90%+ Faster	issue identification and fix, through a test intelligence hub and AI-driven test data analysis
DEPLOY	50%+ Less	test cycle time, through integrated app distribution and beta feedback
MONITOR	75%+ Less	critical crashes, through production error reporting that feeds straight back into the next test cycle

Most importantly: Every step still goes through human review and approval — the difference between a platform you can trust with a release decision and a stack of disconnected tools, each answering 'is this ready to ship?' differently.

Solve the Code Verification Crisis

AI proved that software testing was never built to scale. AURA is what release assurance looks like when it finally does.

Move fast and ship with confidence. See how Sauce Labs closes the loop between business intent, validation, and release confidence.

8.7B
TEST RUNS IN THE DATASET

300K+
ENTERPRISE USERS

80%
OF THE TOP 10 FINANCIAL INSTITUTIONS

Move fast and ship with confidence.

See how Sauce Labs closes the loop between business intent, validation, and release confidence.

BOOK A DEMO

About This Report

ABOUT SAUCE LABS

Sauce Labs is the industry's only full-lifecycle release assurance platform and the company behind Selenium. Trusted by 80% of the world's top 10 largest financial institutions and over 300,000 enterprise users, Sauce Labs provides the only AI platform capable of turning business intent into autonomous testing and quality assurance. With a proprietary dataset of nearly 9 billion test runs, Sauce Labs empowers the Fortune 2000 to bridge the gap between AI-driven code generation and enterprise-grade software quality. Learn more at saucelabs.com.

ABOUT WAKEFIELD RESEARCH

Wakefield Research is the leading market research firm for publicly released data used for earned media and thought leadership. Wakefield also provides research for strategy and insight in nearly 100 countries, and our market intelligence division provides secondary data insights to companies worldwide. Our research has helped more than 50 of the Fortune 100. We specialize in intelligent, practical consultancy with an emphasis on results, the customer experience, and building long-term strategic relationships. Learn more at wakefieldresearch.com.

METHODOLOGY NOTE

The Sauce Labs survey was conducted by Wakefield Research among 200 U.S. executives/C-suites (CTOs, CIOs, CEOs, and similar leaders) and 200 U.S. engineering leaders (directors/associate directors and VPs, SVPs, EVPs of engineering), between May 11 and May 28, 2026, using an email invitation and an online survey. Results of any sample are subject to sampling variation. For the total 400 interviews conducted in this study, the margin of error is ±4.9 percentage points at a 95% confidence level.